

**Experiment 5**

**Date of Performance : 06-03-2023** **Date of Submission: 07-03-2023**

**SAP Id:** 60004200107 **Name :** Kartik Jolapara

**Div:** B **Batch :** B1

**Aim of Experiment**

mplement Hill Cipher. Create two functions Encrypt() and Decrypt(). Demonstrate these ciphers using Color Images / Gray Scale Images

**Theory / Algorithm / Conceptual Description**

Hill cipher is a polygraphic substitution cipher based on linear algebra. Each letter is represented by a number modulo 26. Often the simple scheme A = 0, B = 1, …, Z = 25 is used, but this is not an essential feature of the cipher. To encrypt a message, each block of n letters (considered as an n-component vector) is multiplied by an invertible n × n matrix, against modulus 26. To decrypt the message, each block is multiplied by the inverse of the matrix used for encryption. The matrix used for encryption is the cipher key, and it should be chosen randomly from the set of invertible n × n matrices (modulo 26).

**Program**

|  |
| --- |
| from PIL import Image  from numpy import array  import numpy as np  def modInverse(A, M):  for X in range(1, M):  if (((A % M) \* (X % M)) % M == 1):  return X  return -1  im = Image.open(r"1.jpg")  ar = array(im)  list(ar)  k = int(input("Enter a key value for encryption: "))  key = np.zeros((100,100,3))  for i in range(100):  for j in range(100):  key[i][j] = k  result = np.zeros((100,100,3))  for i in range(100):  for j in range(100):  result[i][j] = (key[i][j] \* ar[i][j]) % 255  print("\nPlain Image")  r = Image.fromarray(np.uint8(ar))  r.show()  print("\nEncrypted Image")  r = Image.fromarray(np.uint8(result))  r.show()  mod = modInverse(k, 255)  key\_inv = np.zeros((100,100,3))  for i in range(100):  for j in range(100):  key\_inv[i][j] = mod  result\_dec = np.zeros((100,100,3))  for i in range(100):  for j in range(100):  result\_dec[i][j] = (key\_inv[i][j] \* result[i][j]) % 255  print("\nDecrypted Image")  r = Image.fromarray(np.uint8(result\_dec))  r.show() |

**Input/Output**

|  |
| --- |
|  |

**CONCLUSION**

Thus, we have successfully implemented Hill cipher